Altera UP1 Board Test

Steven Sutankayo, Noah Aklilu, Duncan Elliott

1.0 Introduction

This document describes the testing procedure for the Altera UP1 boards. The test is useful to students, professors, and TAs who wish to verify that the project boards are working.

2.0 Setting Up The UP1 Board Test

To program the device with a design, the user needs a PC, ByteBlaster or BitBlaster cable, the MaxPlusII software, and the UP1 and power supply.

2.1 Board Jumpers

The UP1 board has four jumpers that control which devices can be programmed. The jumpers must be set correctly, otherwise the board cannot be programmed.

The figures below show the correct jumper settings for configuring or programming the UP1 board.

FIGURE 1. Jumper Settings for Configuring only the EPF10K20 (FLEX) Device

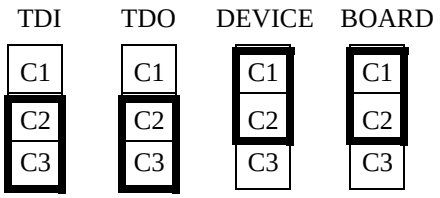


FIGURE 2. Jumper Settings for Programming only the EPM7128S (MAX) Device

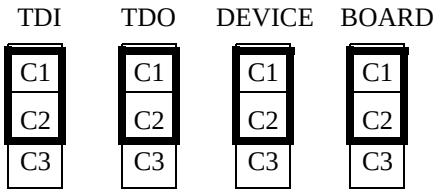
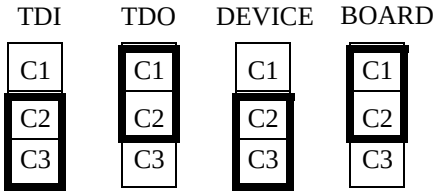


FIGURE 3. Jumper Settings for Configuring/Programming Both Devices



To run the UP1 board test, use the jumper settings shown in figure 3.

2.2 MAXPLUS2 Software

To program the UP1 board the user must be familiar with the use of the Altera MaxPlusII software package.

Before you start, find the programming files for the test. There are two files: max2.pof (for programming the Max chip) and flex2.sof (for configuring the Flex chip).

To program the board, do the following:

1. Start Maxplus II
2. Click on the *Maxplus II* menu and select *Programmer*, the Programmer tool will popup
3. Make sure the *Programmer* tool is the active window and click on the *JTAG* menu.
4. Make sure that there is a check beside *Multi-Device JTAG Chain*
5. Select *Multi-Device JTAG Chain Setup...* to bring up its dialog box

6. if there are any programming files already listed in the box, remove them by selecting them and pressing **Delete**
7. Click on the drop menu of *Device Name* and select *EPM7128S*.
8. click the **Select Programming File** button to bring up a file browsing dialog box
9. browse the directory where the board test files are located, and select the *max2.pof* file, and press **OK**
10. Click **Add** to add your file to the device programming list
11. Click on the drop menu of *Device Name* and select *EPF10K20*.
12. Click the **Select Programming File** button again. This time select the *flex2.sof* test file and press **OK**.
13. Click **Add** to add your file to the device programming list
14. Make sure your project board is powered up, and the jumpers are set properly. Then click on the **Detect JTAG Info** button. It should confirm successfully. If it does not, check that the jumpers are set properly, as described in the previous section. The programming files could also be in the wrong order. Try changing the order in which they appear in the listing.
15. Press **OK**. You will be taken back to the *Programmer* window.
16. Press **Program** to setup the Max chip.
17. Press **Configure** to setup the Flex chip.

The most common thing to go wrong during this operation is to fail when trying to "Detect JTAG Info". Possible causes of this are:

- JTAG cable not connected
- board not powered up
- wrong jumper configuration
- wrong number of files in the device programming list
- files in the device programming list in the wrong order
- software version not capable or licensed for programming (try a different version)

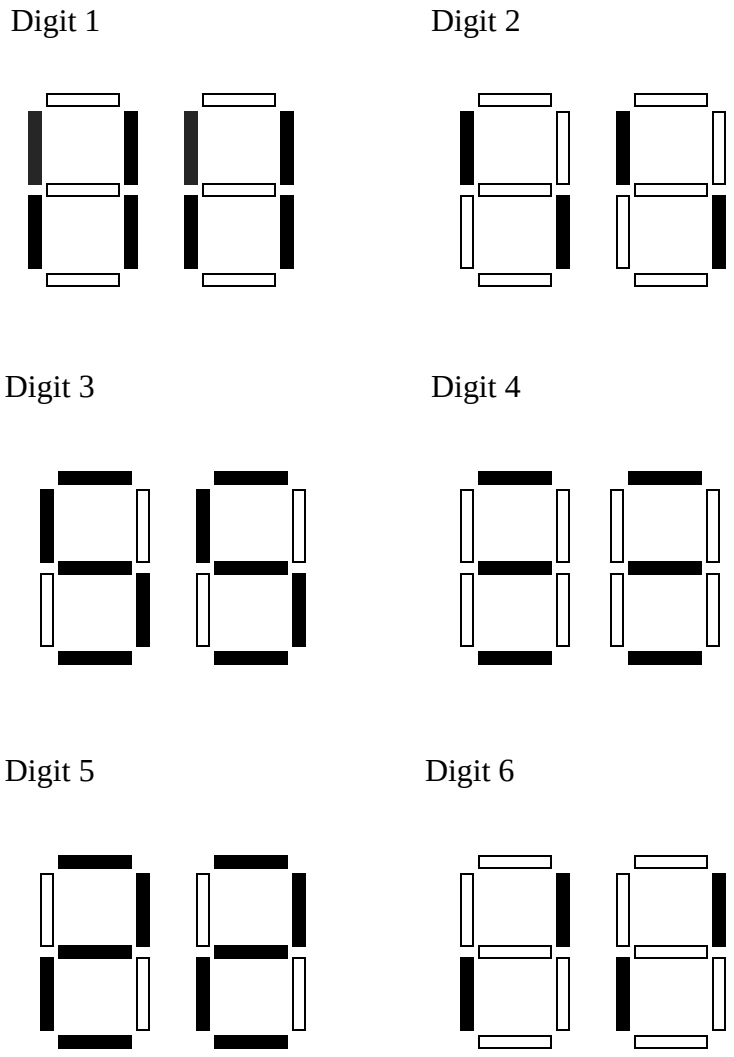
3.0 Test Procedure

If you have not already done so, program the UP1 board devices as described above.

3.1 Seven-segment Display

Both sets of displays (one for the Max chip and one for the Flex chip) should display the following pattern:

FIGURE 4. Pattern of 7-segment display for correct chip operation.



3.2 MAX Test

The pushbuttons on the max chip have no effect on this test, since they have to be manually connected to the prototyping header.

Also, the decimal points on the seven segment display should blink as the digit pattern changes. (don't worry about the exact pattern, though)

3.3 Flex Test

As in the Max test, the dual 7-segment displays should follow the pattern. However, the decimal points should not blink by themselves. The decimal points are connected to pushbuttons and DIP switches.

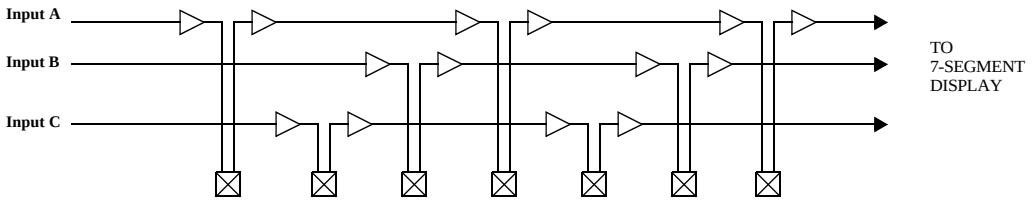
Depress PB1 and PB2 (separately) and verify that the left decimal point blinks. Then toggle all of the DIP switches and verify that the right decimal point changes after each toggle. If they do, the test passes.

4.0 Test Description

Two separate tests have been created to verify that the UP1 board is working properly. One test verifies the Max chip, the other verifies the Flex chip.

Both tests use a three signal input pattern. The signals are routed to all of the I/O pins on the chip. The pins are configured in VHDL as *inout* pins. The pins are assigned sequentially, so that the signals leapfrog over one another:

FIGURE 5. Signal Path in Altera Board Tests



This test architecture allows us to test that all of the I/O pins are working, and that no solder bridges or other short-faults exist between any three adjacent I/O pins.

The input pattern used is shown on the right. The crystal oscillator is used as an input to drive the pattern generator. It is divided down to approximately 2 Hz.

TABLE 4. Test Input Pattern

Input A	Input B	Input C
1	0	0
1	1	0
0	1	0
0	1	1
0	0	1
1	0	1

When the signal is propagated through the chain, it causes the seven segment display to show the pattern described in Figure 4, “Pattern of 7-segment display for correct chip operation.,” on page 4. The order of the pin assignments was manipulated so that both digits of the display showed the same pattern.

4.1 Max Test

The test is designed so that it is not necessary to hook up any wires to the prototyping headers to run the test. Because of this, it’s not possible to test the LEDs or DIP switches in this test (they must be connected throught the prototyping headers manually with hook-up wire).

The test checks that the crystal oscillator is working, the 7-segment display works, that there are no stuck at 0/1 faults on any general-purpose I/O pin, and that there are no solder bridges between any three adjacent I/O pins.

Here is the order of the pin assignments for the I/O signal chain(pins connected to the seven-segment display are in boldface type):

TABLE 4. Pin Assignments for Max Test

	4	5	6	8	9
10	11	12	15	16	17
18	20	21	22	24	25
27	28	29	30	31	33
34	35	36	37	39	40
41	44	45	46	48	49
50	51	52	54	55	56
57	80	81	**58**	**60**	**61**
63	**64**	**65**	**67**	**68**	79
69	**70**	**73**	**74**	**76**	**75**
77					

4.2 Flex Test

The Flex test is basically the same as the Max test, with a few changes.

The Flex test does not test 100% of the general purpose I/O pins. Some of the pins were not used for the test because they are connected to the mouse and VGA ports on the board. These pins are configured as “reserve pins” in the project file.

The decimal points are connected to switches. The left decimal point is connected to the XOR of PB1 and PB2. The right decimal point is connected to the XOR of all 8 DIP switches.

The following page shows the order of the pin assignments for the I/O signal chain(pins connected to the seven-segment display are in boldface type):

TABLE 4. Pin Assignments for Flex Test

43	44	45	46	48	49
50	51	53	54	55	56
61	62	63	64	65	66
67	68	70	71	72	73
74	75	76	78	79	80
81	82	83	84	86	87
88	94	95	97	98	99
100	101	102	103	105	106
107	108	109	110	111	113
114	115	116	117	118	119
120	126	127	128	129	131
132	133	134	136	137	138
139	141	142	143	144	146
147	148	149	151	152	153
154	156	157	158	159	161
162	163	164	166	167	168
169	171	172	173	174	175
181	182	183	184	185	186
187	188	190	191	192	193
194	195	196	198	199	200
201	202	203	204	206	207
208	209	213	214	217	218
219	220	221	222	223	225
226	228	229	230	231	**6**
7	**8**	**9**	**11**	**12**	**13**
15	26	**17**	**18**	**19**	**20**
21	**23**	**24**	233		

5.0 Project Files

Max test VHDL source file	max2.vhd
Max test programming file	max2.pof
Flex test VHDL source file	flex2.vhd
Flex test programming file	flex2.sof